

Guía de ejercicios evaluables → no se suben soluciones

3 disciplinas sobre las que func. la Ing. en soft. (3P)

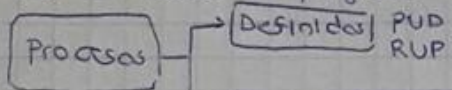
Proceso la más abstracta y producto, el más concreto.

(incluye planificación, organización, seguimiento)

Proyecto: es una unidad de gestión y organización de recursos y personas para obtener como resultado un producto o servicio.

Rol de Ing. de procesos:

Rol de líderes de proyecto



PUD  
RUP

Definición de un proceso:

Intención de considerar todas las cosas necesarias para hacer soft. (artef. entendi., roles, roles, métricas)

2. Buena experiencia

Empodera a las equipos de proyecto

→ Ágil

→ Lean (viene de la ind. automotriz)

2. Corriendo ahora están mezclados

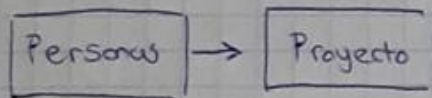
3. Sin pensamiento o filosofías, no metodología

Procesos definidos buscan repetibilidad para inferir.

Personas son el elemento fundamental para hacer software

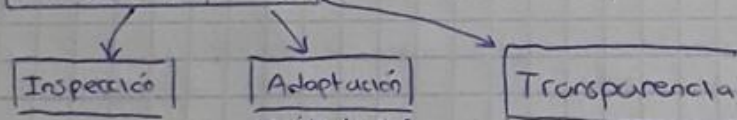
El mayor costo (80% esfuerzo)

hs. persona lineales



Problemas de comunicación para empezar en la toma de requerimientos

Procesos empíricos: se basan en 3 pilares



corrige lo que se necesita

Como neces. aprender de la experiencia neces. exp. rápido. ciclos cortos.

competencia que debemos desarrollar: manejo de info. relacionarse con el cliente y usuarios.

Se debe trabajar con ciclos iterativos si o si (no puede ser cascada)

No importa cual ciclo iterativo pero debe ser iterativo.

Mientras más corta la iteración mejor. Porque nos damos cuenta antes del error.

Manifiesto Ágil → compuesto por principios y valores

manifestoagil.org (lo vamos a tener que leer)

Scrum habla de como gestionar Proyecto no de construir productos.

Infografía → se priorizan las funciones.

El equipo de proyecto se compone por gente de negocio y gente técnica

Product Owner está del lado del negocio: es el que tiene las necesidades, forma parte (esto no se consideraba en los procesos definidos).

Debe tener capacidad de toma de decisiones para que sea útil.

Aceptar que los req. cambian

Ir y preguntarle al cliente, 50% de los req aparecen cuando el cliente ve el soft.

Empoderamiento de las miembros del equipo

Individuos motivados, capacitados y con autogestión, también la auto-organización.

Darle al equipo cap<sup>ta</sup> el  
ciudad de toma de decisiones

Las mejores ideas salen del equipo

El mejor método de comunicación es cara a cara

Importancia de tener equipos co-locados (en la misma ubicación física)

Si no puede estar que llegue tras llamarlo a que el cliente  
poco tiempo.

Ágil

Su manifiesto funciona como un paraguas que cubre a muchos frameworks  
valoramos hacer entregas frecuentes de lo que es de valor para el cliente

Requerimientos Ágiles

El software no es un fin en si mismo es una herramienta para que nuestros clientes  
tengan valor de negocio.

Conocer el negocio para poder construir el producto de software correcto.

Mike Cohn → referente de User Stories →

Definir lo que es suficiente. Vamos a entregar poca funcionalidad cada vez

45% de las func de un prod de soft no se usa nunca

Hacer foco en entregar MVP (producto mínimo viable)

El porcentaje de func que si son necesarias

Trabajar haciendo un producto que no está completamente definido

Product Backlog: lista priorizada de características del producto.

Están ordenadas por lo que el product owner necesita antes que lo otro.

El dueño del product Backlog es el product owner, puede hacer cambios cuando  
quiera. los req emergentes van cayendo ahí y se van sucando las que ya se trabajaron

Just in time

No invertir tiempo en espectf de req que no vamos a necesitar.

Hay que analizarlos cuando se los necesita.

Eliminar desperdicios

evitar

BRUF invertir tiempo en hacer grandes req cuando el 80% no sirve.

Los requerim tienen 21st grados de granularidad.

User stories → hablamos de ~~reg de usuario~~ <sup>son</sup> reg de usuario. <sup>realiza de CU</sup> reg de soft

o las debe escribir el product owner (por lo menos la idea debe venir de él, se lo  
puede asesorar).

3 variables que considerar y regular:

- Alcance
- Costos
- Tiempo



definido  $\rightarrow$  dirigido por un plan  
Una vez definidos los req no cambian los req y tiempo se estiman.

Agil o dirigido por valores:

Define una iteración de tiempo fija (ej un mes) ahí se acuerda juntos que puede entregarse. Vienen a estar permeables a cambiar.

Principios son los que fundamentan los user stories.

User stories - descripción corta de una func definida para el usuario

No existen las batas de plata  $\rightarrow$  lo más difícil es decidir qué soft hacer.  
(no sirve Bullet)  $\rightarrow$  la pregunta en el técnico sobre este artefacto.

Habla de las caract esenciales y accidentales del soft.

Partes de una user story



$\rightarrow$  rol  
Quién, qué, para qué

Es importante respetar la sintaxis de la user stories

Los usa el proyecto a los user stories

Hay algunas cuya granularidad es más gruesa (épicas o temas)  $\rightarrow$  No entra en una iteración  $\rightarrow$  agrupaciones de users o épicas  
Prioridad baja.  
Arrriba en el product backlog siempre va a haber user stories.

Parcelas verticales - De incluir un pedacito de cada cosa.

Criterios de aceptación

Son límites que pongo sobre los users para que no queden dudas en algunas requerimientos

Son los que nos ayudan a armar las pruebas de aceptación.

Requerimientos Ágiles

22/08

Cuando veremos los cu especificabamos los req

En los user stories pienso el como recién cuando lo voy a programar. Más que nada lo que buscamos es identificar  
Apuntamos a operar o el último momento para especificar.

Una user ~~story~~ story es un token para una <sup>conversación</sup> ~~funcionalidad~~. Es una conversación con el usuario.

El equipo define el template de la user story

Servicio para definir los requisitos

casos que tienen que pasar para que el usuario acepte (reglas de negocio esperadas vinculadas a este user y ERF)

Los debe aceptar el Product Owner (ni cliente, ni el usuario)  
El cliente es el que pone la plata  
Lo importante es que sepa del negocio

Frase Verbal
Como [rol] quiero/puedo [acción/que] para [valor de negocio]
Criterios de Aceptación [...]
Pruebas de Aceptación (con como elemento de rol)

- Exito
- Falla

### Festival de Folklore

Los historias tienen que ser independientes  
Mejorarse es posible de poner el par de datos

Vender Entradas (para el negocio)
Como Vendedor quiero vender las entradas de un festival para que los clientes puedan ingresar
Criterios de Aceptación
<ul style="list-style-type: none"> <li>• venta unicamente en efectivo</li> <li>• el número de entradas vendidas debe ser mayor a cero y menor al número de entradas disp</li> </ul>
Pruebas de aceptación
<ul style="list-style-type: none"> <li>• Probar vender entradas cuando no hay disp (falla)</li> <li>•             hay disp (pasa)</li> <li>•          ingresando un monto de plata mayor al total (pa)</li> <li>•                         menor       (falla)</li> <li>• Probar que se genere la entrada en 6 seg (pasa) (se si podría pasar en 1 seg)</li> </ul>

Situaciones de falla (situaciones en que el soft funci bien cuando debería fallar).  
Que no pase algo que no debería pasar

Aplicación  
Mis Gastos  
Familiares

- Iniciar sesión
- Registrar gastos
- Administrar tipo de gasto (soporte)
- Registrar usuario
- Consultar gastos (o visualizar)
- Filtrar gastos
- Cerrar sesión
- Administrar responsable de gasto

Una iteración dura lo que dura un equipo (entre 1 y 4 semanas)  
Aceptación de la gestión  
Aceptación completa

## Registrar Gasto

Como Responsable de gastos quiero registrar un gasto para una determinada fecha para poder gestionar los gastos de mi familia.

### Criterios de aceptación

- La fecha del gasto debe ser anterior o igual a la actual
- Se debe ingresar un monto positivo
- Se puede cambiar de responsable
- Se debe especificar el tipo de gasto
- Se debe poder filtrar fechas y personas mostrando coincidencias

### Pruebas de Aceptación

- Probar ingresar un monto positivo al registrar un gasto (para)
- Probar ingresar un monto negativo al registrar un gasto (falla)
- Probar ingresar una fecha posterior de la actual (falla)
- Probar ingresar una fecha anterior (para)
- Probar cambiar el responsable de gasto (para)
- Probar ingresar primero 3 veces y me muestra responsable de gasto (para)
- Probar ingresar sin monto (falla)

12/09 Primer ejercicio evaluable  
Implementación de User.

25/08

### Requisitos ágiles

(El soft es un medio para un fin, no es el fin en si mismo)

Entregando por:

→ Entrega de valor de negocio

→ Entrega frecuente de software funcionando

→ El producto no está 100% definido

Difícil que el cliente acepte iniciar el proyecto con una visión global y algunos req.

↓  
visión  
↓  
Product Backlog

(tener idea o 100% de los req. va de acuerdo a lo que quiere)

→ Eliminar Desperdicios.

→ Diferrir las decisiones

Just in time

(veremos si especificar lo que necesitamos lo que necesitamos)

En el contexto de los req. Ágiles surgen los User stories.

Estimar → complejo

→ universo de incertidumbre

2 principios de Lean que se tomaron sin

- Elim desperdicios

(se perdieron mucho

tiempo en espec de

automatización, alto riesgo

de que después cambiar

de opinión).

- Diferrir las decisiones

hasta el último momento responsable

Postergan hasta tener un

poco más de información

para la toma de decis.

↓  
Cuando todavía se puede decidir.

